

FIG. 1

201 DATA_TYPE = DATA TYPE NAME + [PARENT DATA TYPE REF] + [(ELEMENT)*]

202 DATA_TYPE_NAME = a name that uniquely identifies this datatype from other datatypes

203 PARENT_DATA_TYPE_REF = a reference to another datatype using it's DATA TYPE NAME. This value indicates that this datatype is a "descendent" of PARENT DATA TYPE REF

204 ELEMENT = ELEMENT NAME + [DATA TYPE REF] + [POSITIONAL REFERENCE] + [ALIAS NAME] + [(ELEMENT)*]

205 ELEMENT_NAME = a name that identifies this element

206 DATA_TYPE_REF = a reference to another datatype using it's DATA TYPE NAME. This value indicates that child structure of this element is at least equal to the child structure of the datatype referenced.

207 POSITIONAL_REFERENCE = ELEMENT REF

208 ELEMENT_REF = a reference to a child element of the datatype specified by DATA TYPE REF in this elements parent element.

209 ALIAS_NAME = a reference to a child element of the datatype specified by DATA TYPE REF in this elements parent element. When specifying this value, it indicates that the element referred to by ALIAS NAME is now replaced by ELEMENT NAME

FIG. 2

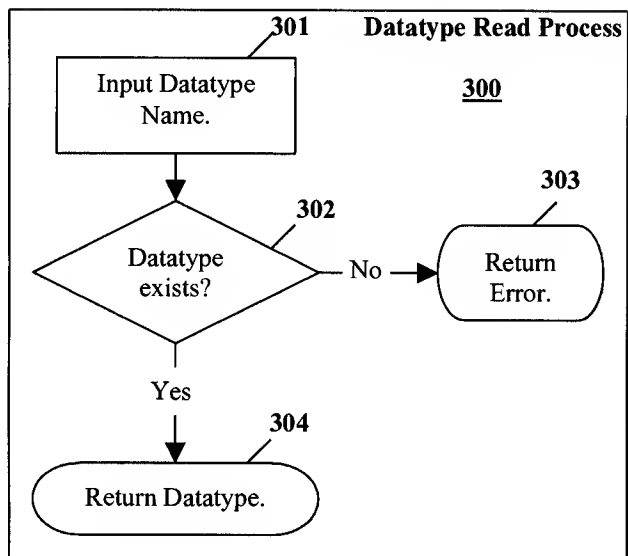


FIG. 3A

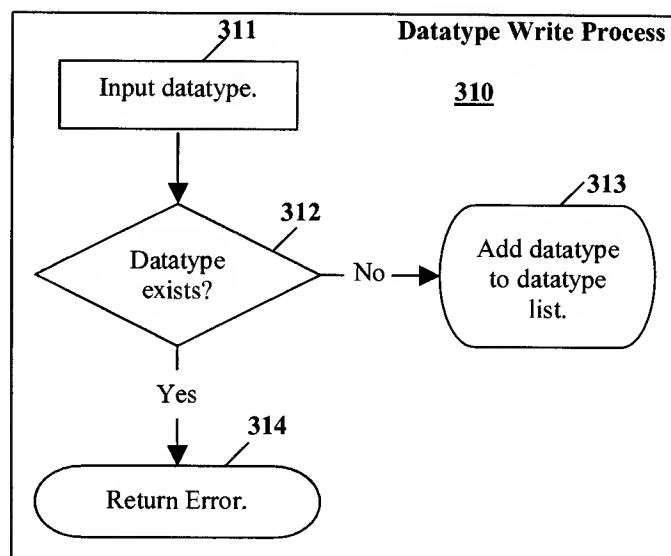


FIG. 3B

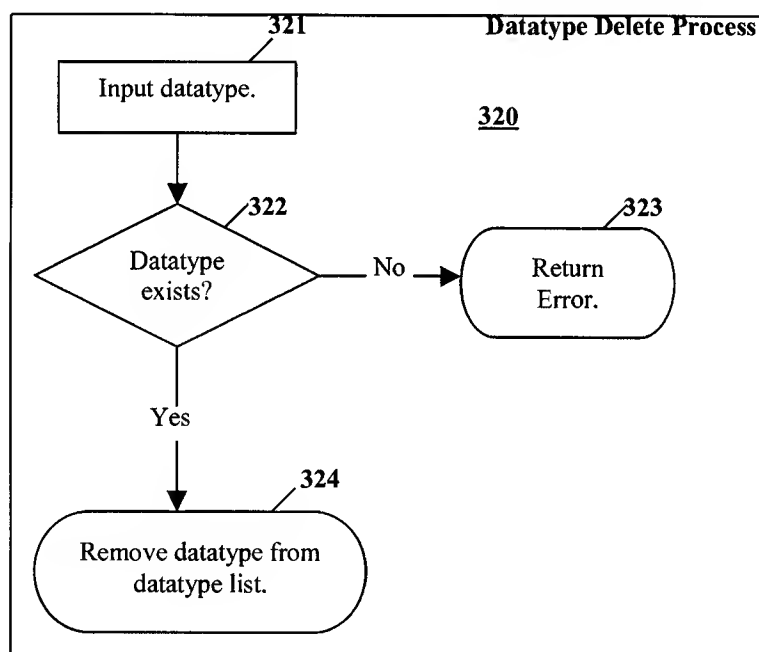


FIG. 3C

Locate Process #1: Common Family

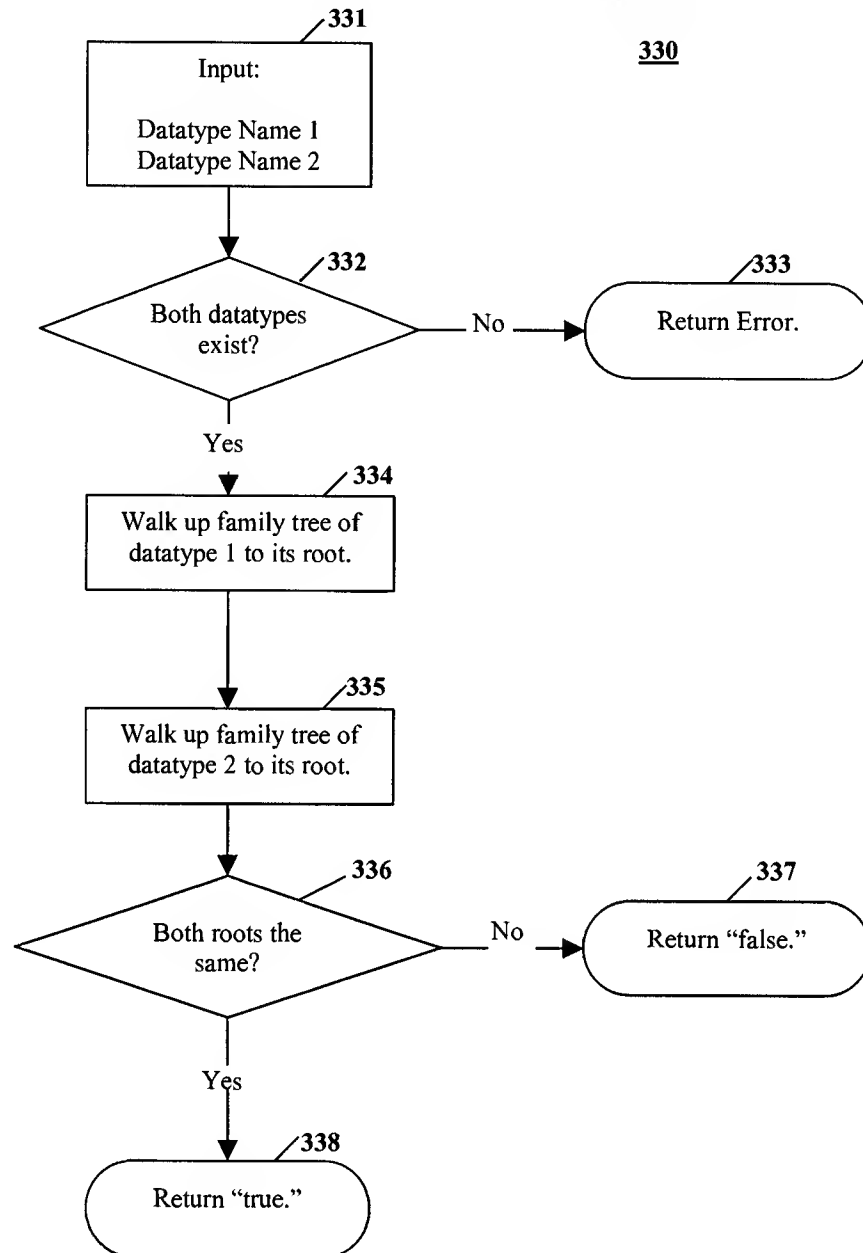


FIG. 3D

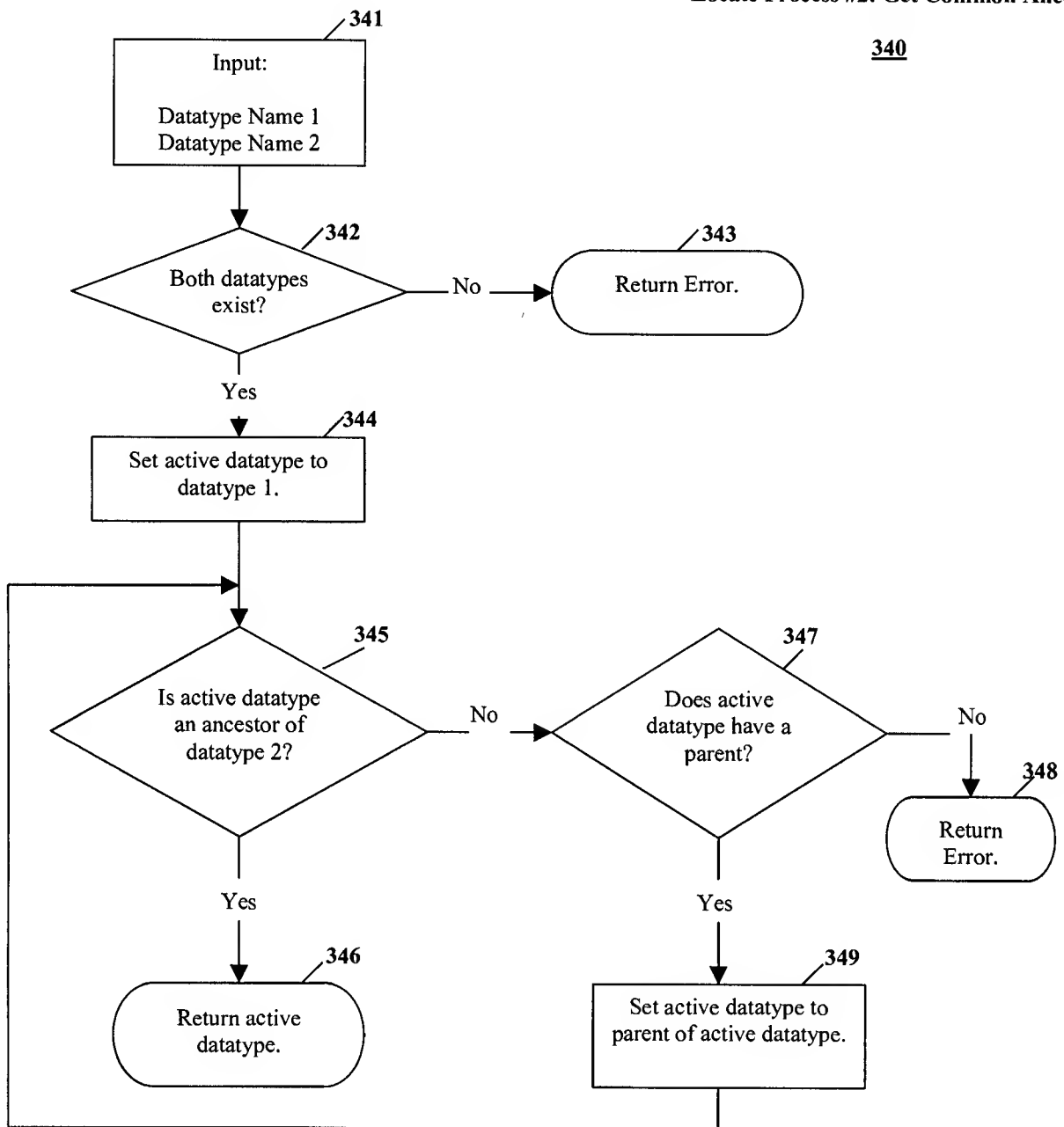
340

FIG. 3E

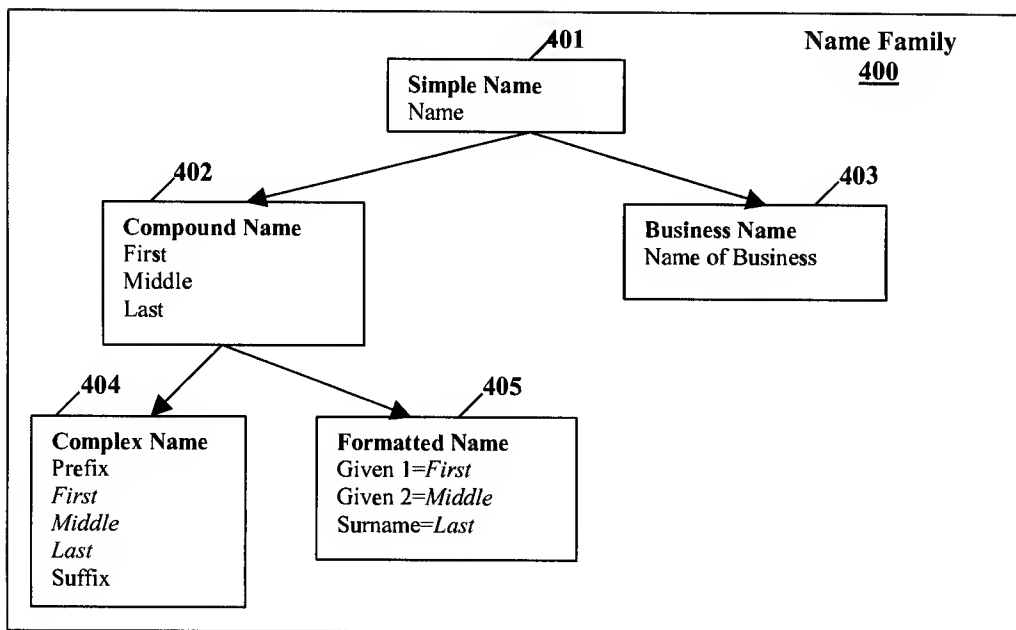


FIG. 4A

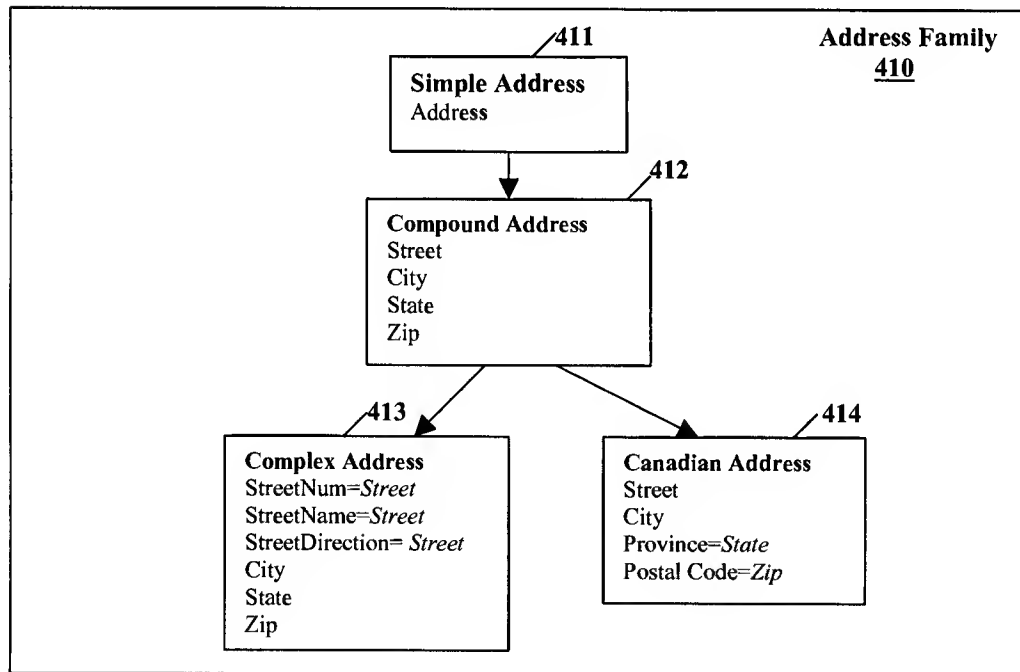


FIG. 4B

```

<SimpleName/>

<CompoundName instanceOf="SimpleName">
    <First/>
    <Middle/>
    <Last/>
</CompoundName>

<ComplexName instanceOf="CompoundName">
    <Prefix insert="First"/>
    <Suffix/>
</ComplexName>

<FormattedName instanceOf="CompoundName">
    <Given1 alias="First"/>
    <Given2 alias="Middle"/>
    <Surname alias="Last"/>
</FormattedName>

<BusinessName instanceOf="SimpleName"/>

```

500

FIG. 5A

```

<Person>
    <Name instanceOf="CompoundName">
        <First/>
        <Middle/>
        <Last/>
    </Name>
    <Address instanceOf="CompoundAddress">
        <Street/>
        <City/>
        <State/>
        <Zip/>
    </Address>
    <DateOfBith instanceOf="Date"/>
    <SSN/>
</Person>

```

510

FIG. 5B

Schema A

```
<CustomerInfo>
  <Name instanceOf="CompoundName">
    <First/>
    <Middle/>
    <Last/>
  </Name>
  <Address instanceOf="CompoundAddress">
    <Street/>
    <City/>
    <State/>
    <Zip/>
  </Address>
  <Race/>
  <Gender/>
  <DOB/>
  <Income/>
  <CreditInfo>
    <Type/>
    <Number/>
    <ExpirationDate/>
  </CreditInfo>
</CustomerInfo>
```

600

Schema B

```
<Invoice>
  <Purchaser>
    <Name instanceOf="ComplexName">
      <First/>
      <Middle/>
      <Last/>
    </Name>
    <Address instanceOf="CanadianAddress">
      <Street/>
      <City/>
      <Province/>
      <PostalCode/>
    </Address>
    <CreditCard instanceOf="CreditCard">
      <Type/>
      <Name/>
      <Number/>
      <ExpiryDate/>
    </CreditCard>
  </Purchaser>
  <Product>
    <SKU/>
    <Description/>
    <Price/>
  </Product>
</Invoice>
```

FIG. 6

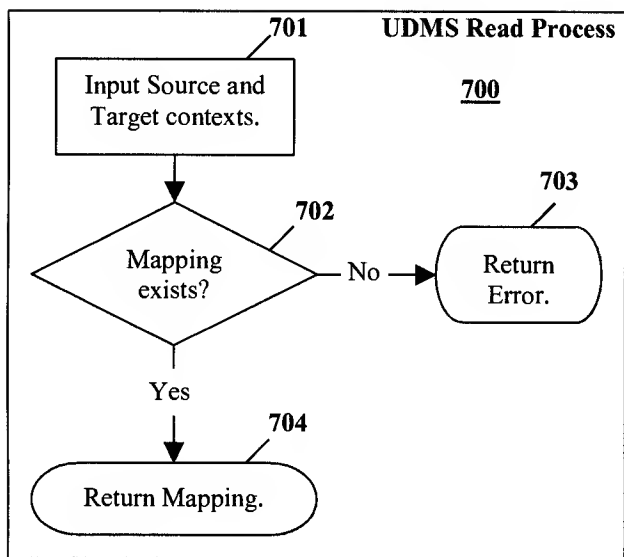


FIG. 7A

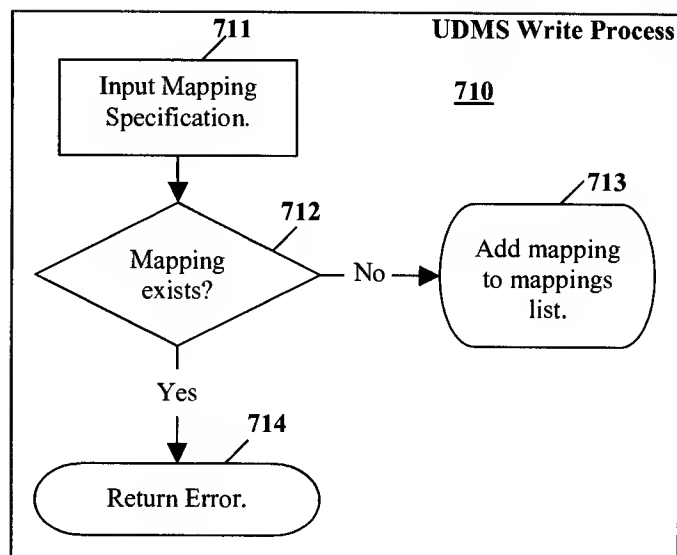


FIG. 7B

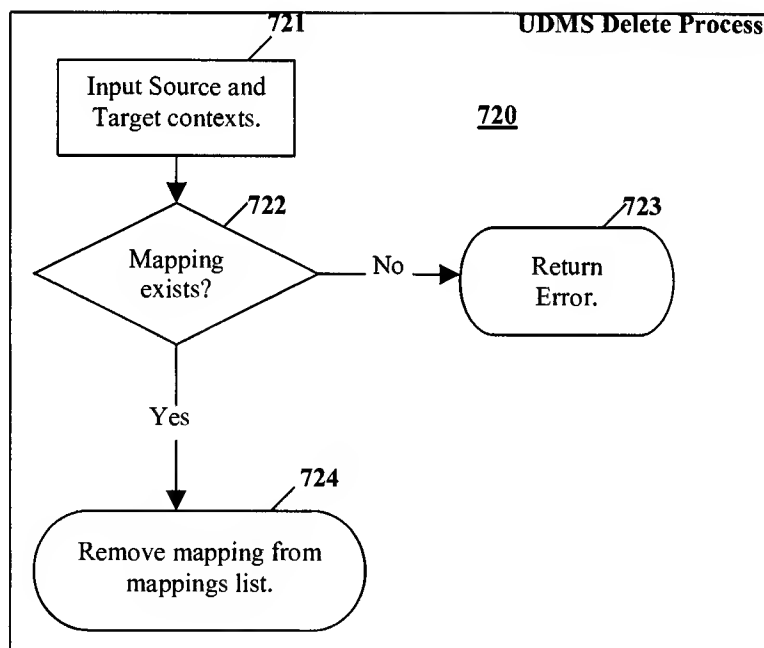


FIG. 7C

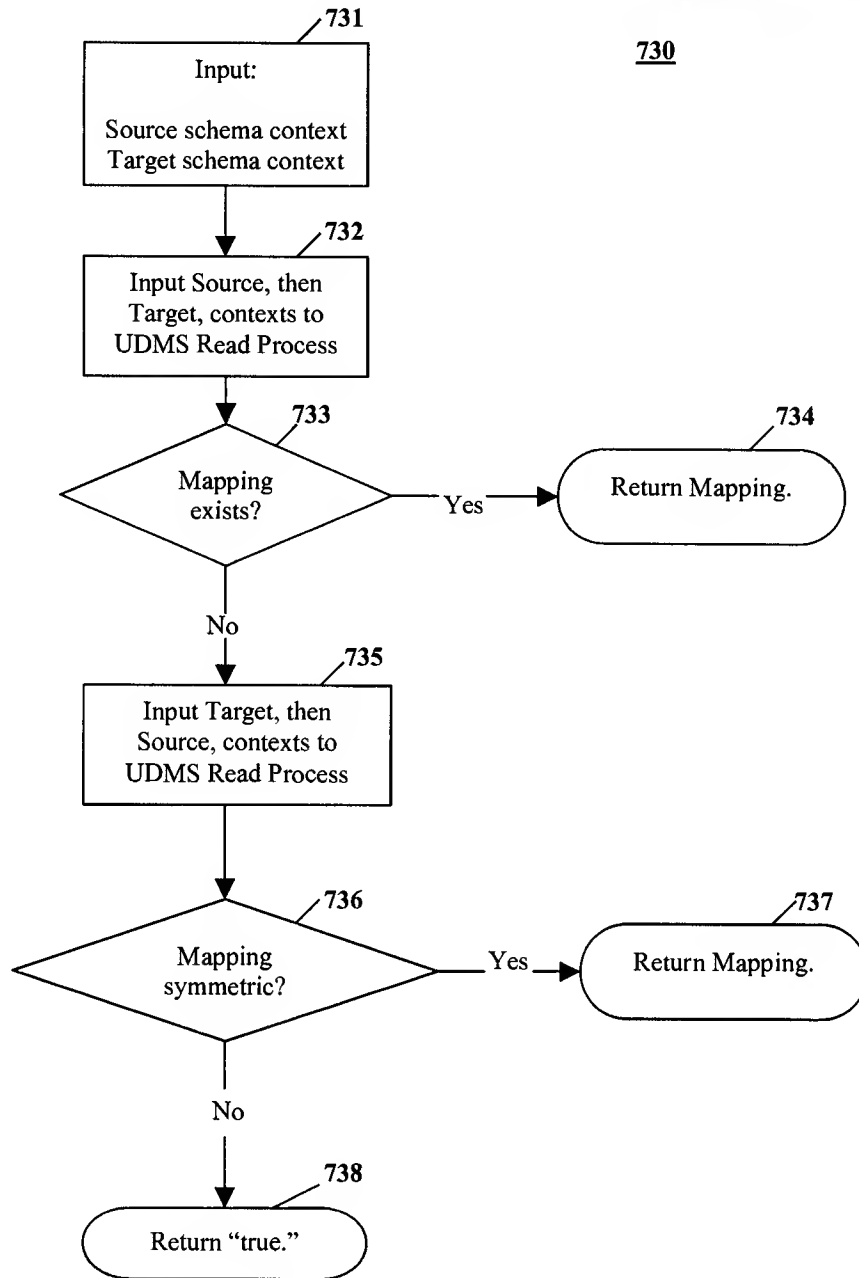
730

FIG. 7D

801 $\text{CONTEXT_MAP} = \text{SYMMETRIC_DESIGNATION} + \text{SOURCE} + \text{TARGET}$

802 $\text{SYMMETRIC_DESIGNATION} = \text{indicates whether a TARGET} \Rightarrow \text{SOURCE mapping is also implied}$

803 $\text{SOURCE} = \text{SCHEMA CONTEXT}$

804 $\text{TARGET} = \text{SCHEMA CONTEXT}$ 800

805 $\text{SCHEMA_CONTEXT} = \text{PARENT ELEMENT CONTEXT} + \text{DELIMITER} + \text{ELEMENT NAME}$

806 $\text{PARENT_ELEMENT_CONTEXT} = \text{SCHEMA CONTEXT of the element's parent (if one exists)}$

807 $\text{DELIMITER} = \text{some known character value that doesn't appear in the ELEMENT NAMES that make up this context}$

FIG. 8A

```
<Map symmetric="true">
  <Source>CustomerInfo</Source>
  <Target>Invoice/Purchaser</Target>
</Map>

<Map symmetric="false">
  <Source>Incident/Suspect</Source>
  <Target>AuctionCompany/Auctions/Seller</Target>
</Map>
```

810

FIG. 8B

- 901 **COMPARISON_ALGORITHM** = ALGORITHM_NAME + IMPLEMENTATION_REFERENCE [+
IMPLEMENTATION_PARAMETERS]
- 902 **ALGORITHM_NAME** = a unique way to identify this algorithm from other
algorithms.
- 903 **IMPLEMENTATION_REFERENCE** = a means to identify an implementation of this
algorithm. This may include, but is not limited to class name, function
call names, and dynamically loadable libraries.
- 904 **IMPLEMENTATION_PARAMETERS** = a set of parameters used to configure this
specific instance of implementation

FIG. 9A

```

<STRING_DIFFERENCE class="com.company.comparisons.StringDiffScore"/>

<SOUNDEX class=" com.company.comparisons.SoundexScore"/>

<NAME_SYNONYM class="com.company.comparisons.SynonymScore">
  <SIMILAR degree="0.9">
    <ELEMENT>Robert</ELEMENT>
    <ELEMENT>Bob</ELEMENT>
    <ELEMENT>Rob</ELEMENT>
    <ELEMENT>Bobby</ELEMENT>
    <ELEMENT>Robby<ELEMENT>
  </SIMILAR>
  <SIMILAR degree="0.85">
    <ELEMENT>John</ELEMENT>
    <ELEMENT>Johnny</ELEMENT>
    <ELEMENT>Jon</ELEMENT>
    <ELEMENT>Juan</ELEMENT>
    <ELEMENT>Jack<ELEMENT>
  </SIMILAR>
</SYNONYM>

```

910

FIG. 9B

**TABLE OF COMPARISON TYPES
USED IN STRATEGIES OF TTE**

	<u>Inputs Received</u>	<u>Success Indicators</u>
Context Comparison	Source schema context. Target schema context.	Existence of mapping specification (including any symmetric versions) is found, using UDMS.
Element Comparison	Two element names. Name Comparison Algorithm. Normalized threshold score.	Calling the Name Comparison Algorithm with the two element names results in a normalized score equal to or greater than the threshold score.
Attribute Comparison	Two attribute values. Attribute Comparison Algorithm. Normalized threshold score.	Calling the Attribute Comparison Algorithm with the two attribute values results in a normalized score equal to or greater than the threshold score.
Datatype Lineage Comparison	Two Datatype Names. Reference to Lineage Comparison Algorithm that is registered with SSS. Normalized threshold score.	Calling the Lineage Comparison Algorithm with the two Datatype Names results in a normalized score equal to or greater than the threshold score.
Datatype Tree/Structure Comparison	Two hierarchical data structures. Reference to Tree Comparison Algorithm that is registered with SSS. Normalized threshold score.	Calling the Tree Comparison Algorithm with the two hierarchical data structures results in a normalized score equal to or greater than the threshold score.

FIG. 10

1101 *TTE = (STRATEGY) **

1102 *STRATEGY = (COMPARISON_TYPE) **

1103 *COMPARISON_TYPE = CONTEXT_COMPARE | ELEMENT_COMPARE | DATATYPE_COMPARE |*
ATTRIBUTE_COMPARE

1104 *CONTEXT_COMPARE = determines if a map exists in the User-Defined Mapping Services*
for two SCHEMA CONTEXTs (including a symmetric version).

1105 *ELEMENT_COMPARE = NAME_COMPARISON_ALGORITHM + THRESHOLD*

1106 *ATTRIBUTE_COMPARE = ATTRIBUTE_NAME + NAME_COMPARISON_ALGORITHM + THRESHOLD*

1107 *NAME_COMPARISON_ALGORITHM = a comparison algorithm registered in the Similarity*
Scoring Services that compares two ELEMENTS_NAMES or two ATTRIBUTE_VALUES and
returns a normalized score.

1108 *DATATYPE_COMPARE = LINEAGE_COMPARE | CHILD_STRUCTURE_COMPARE*

1109 *LINEAGE_COMPARE = LINEAGE_COMPARISON_ALGORITHM + THRESHOLD*

1110 *LINEAGE_COMPARISON_ALGORITHM = a comparison algorithm registered in the*
Similarity Scoring Services that compares datatypes and returns a normalized
score that indicates proximity of the data types are in their family tree.

1111 *CHILD_STRUCTURE_COMPARE = TREE_COMPARISON_ALGORITHM + THRESHOLD*

1112 *TREE_COMPARISON_ALGORITHM = a comparison algorithm registered in the Similarity*
Scoring Services that compares two data hierarchies and returns a normalized
score based on the similarity of their child structures.

1113 *THRESHOLD = a normalized score indicating similarity or proximity.*

FIG. 11

```

<TTE>
  <STRATEGY>
    <MAP/>
  </STRATEGY>
  <STRATEGY>
    <ELEMENT compare="exact" threshold="1.0"/>
    <DATATYPE compare="lineage" threshold="1.0"/>
    <ATTRIBUTE value="description" compare="exact" threshold="1.0"/>
  </STRATEGY>
  <STRATEGY>
    <ELEMENT compare="exact" threshold="1.0"/>
    <DATATYPE compare="lineage" threshold="1.0"/>
    <ATTRIBUTE value="description" compare="string_diff" threshold="0.8"/>
  </STRATEGY>
  <STRATEGY>
    <ELEMENT compare="exact" threshold="1.0"/>
    <DATATYPE compare="lineage" threshold="1.0"/>
  </STRATEGY>
  <STRATEGY>
    <ELEMENT compare="exact" threshold="1.0"/>
    <DATATYPE compare="lineage" threshold="0.5"/>
    <ATTRIBUTE value="description" compare="exact" threshold="1.0"/>
  </STRATEGY>
  <STRATEGY>
    <ELEMENT compare="exact" threshold="1.0"/>
    <DATATYPE compare="lineage" threshold="0.5"/>
    <ATTRIBUTE value="description" compare="string_diff" threshold="0.8"/>
  </STRATEGY>
  <STRATEGY>
    <ELEMENT compare="exact" threshold="1.0"/>
    <DATATYPE compare="lineage" threshold="0.5"/>
  </STRATEGY>
  <STRATEGY>
    <ELEMENT compare="string_diff" threshold="1.0"/>
    <DATATYPE compare="structure" threshold="1.0"/>
    <ATTRIBUTE value="description" compare="string_diff" threshold="0.8"/>
  </STRATEGY>
</TTE>

```

FIG. 12

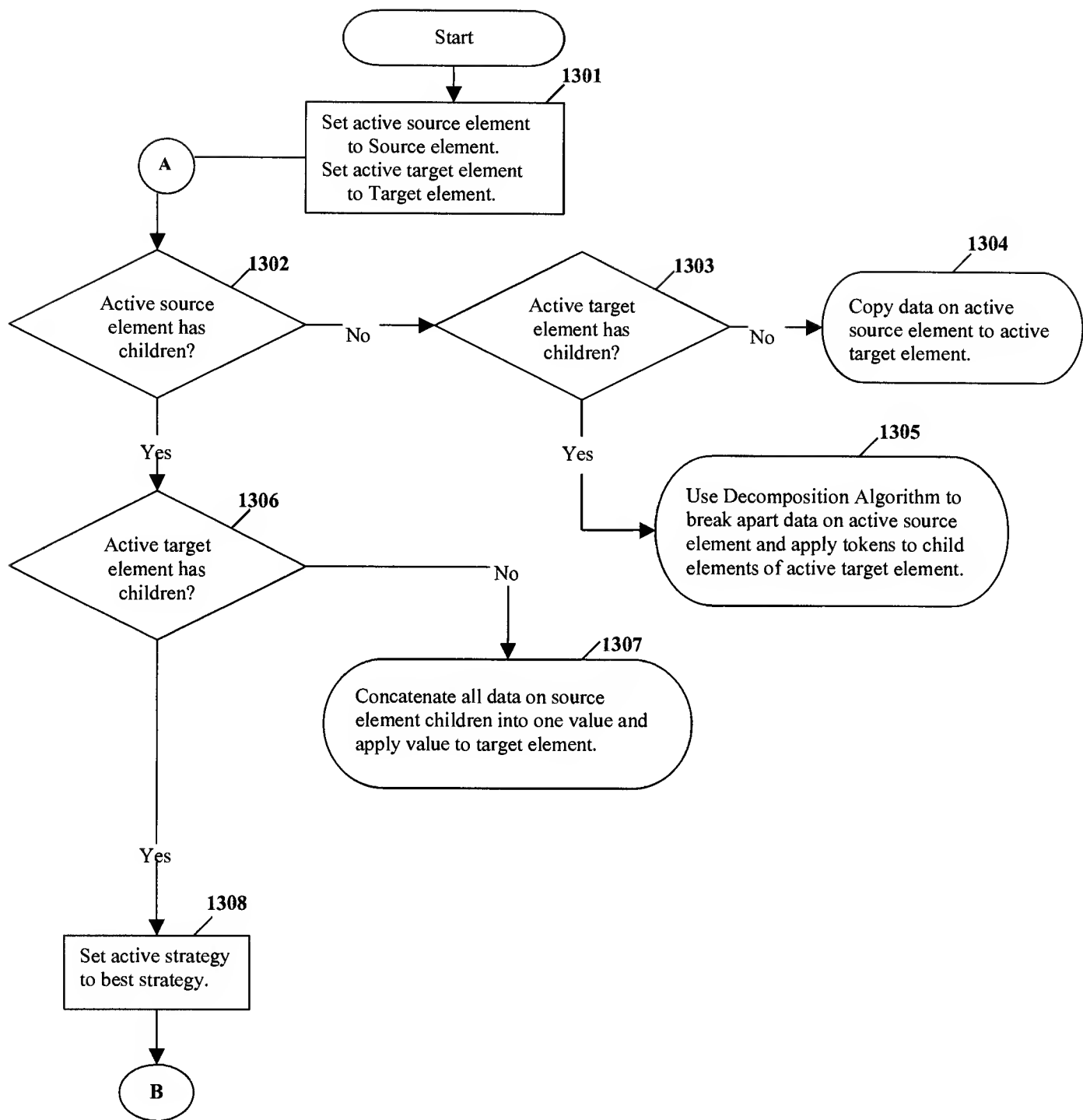


FIG. 13A

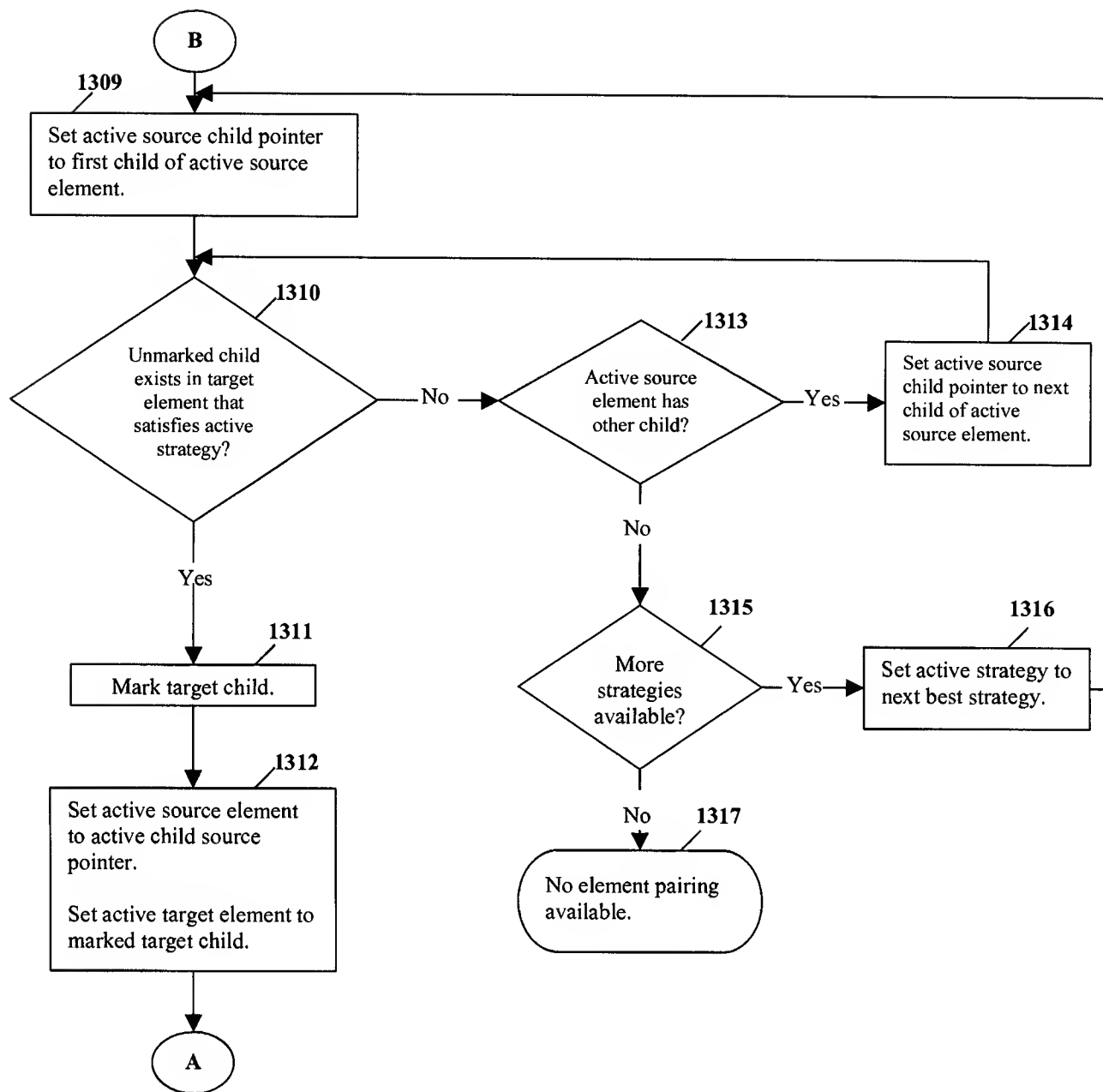


FIG. 13B

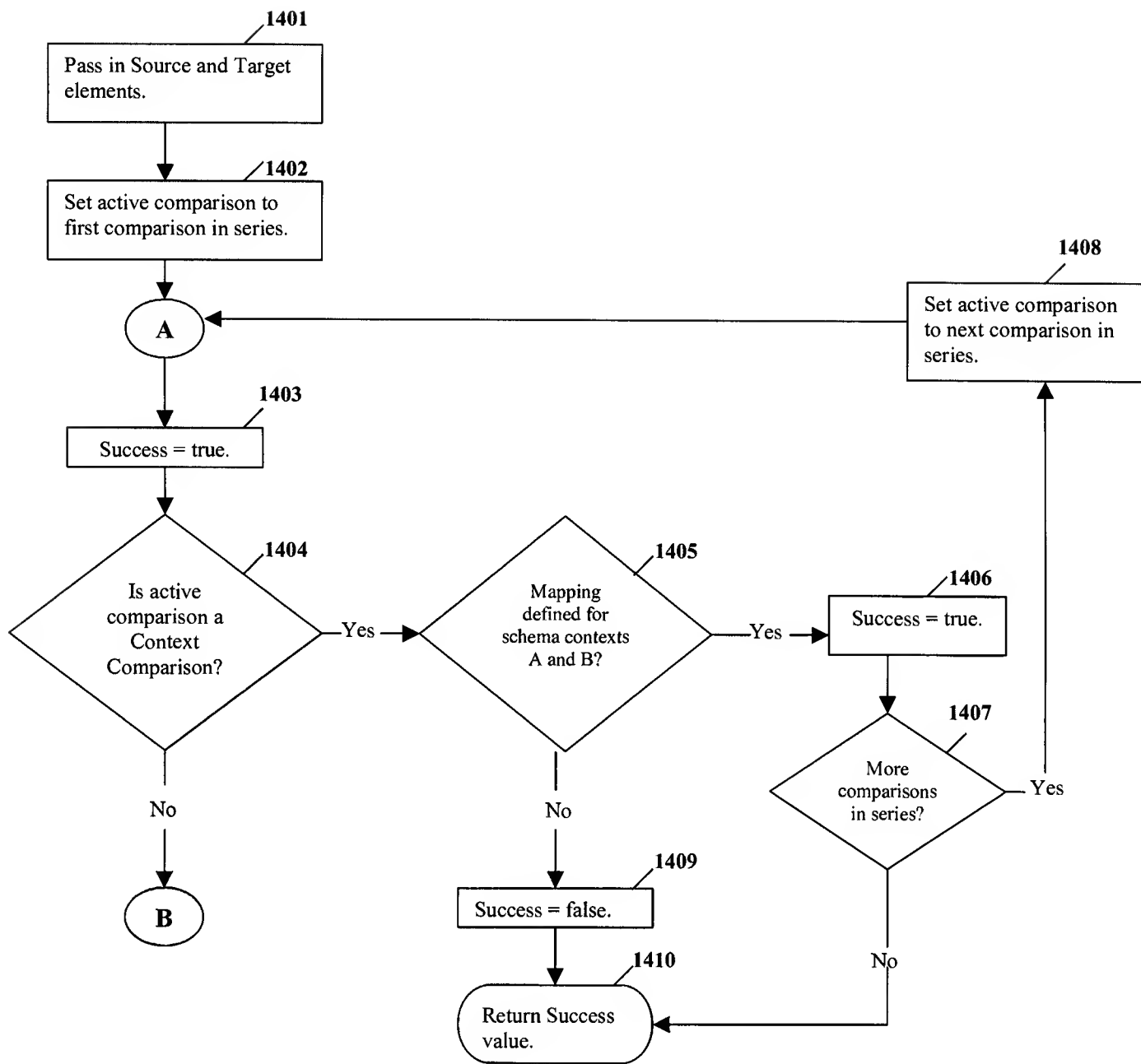


FIG. 14A

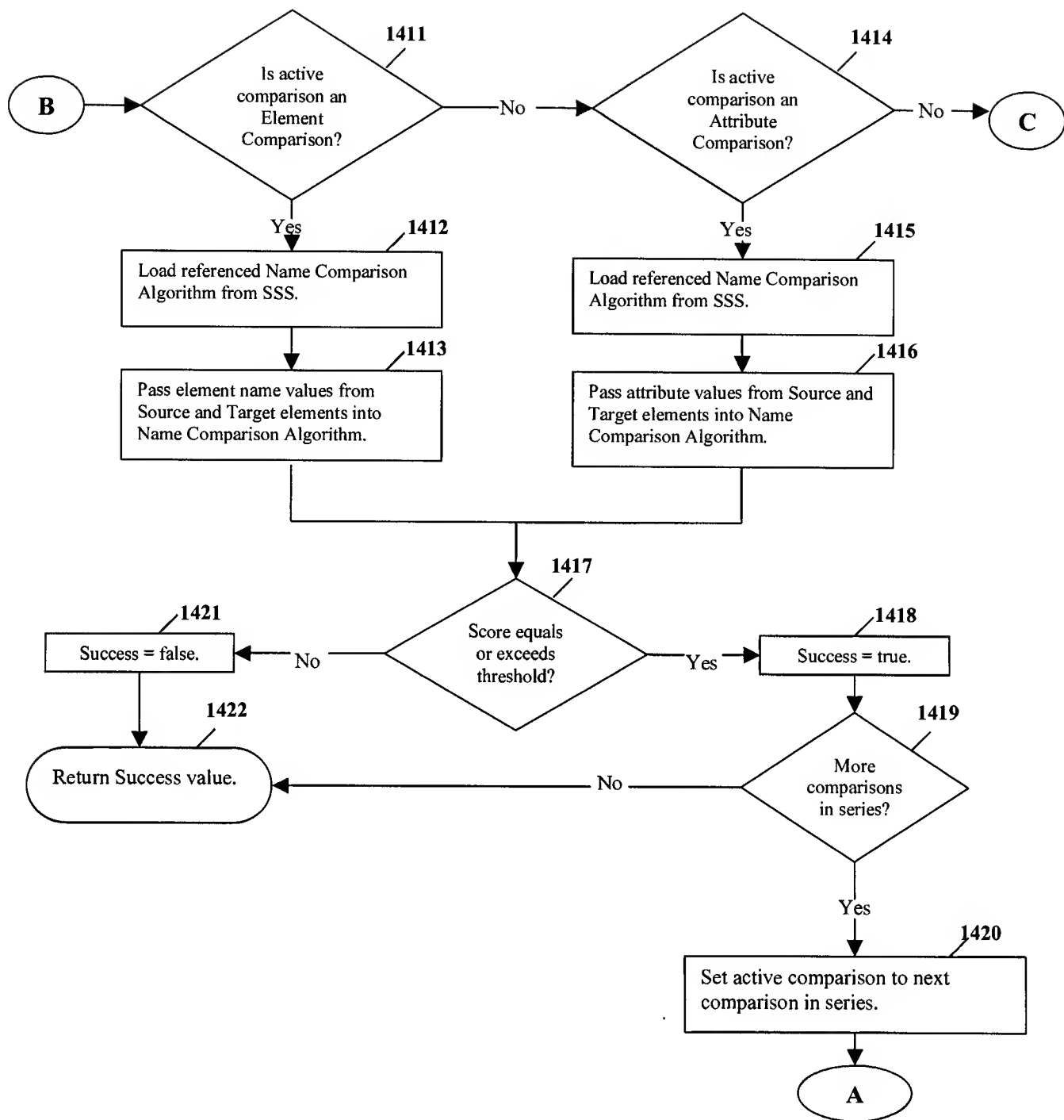


FIG. 14B

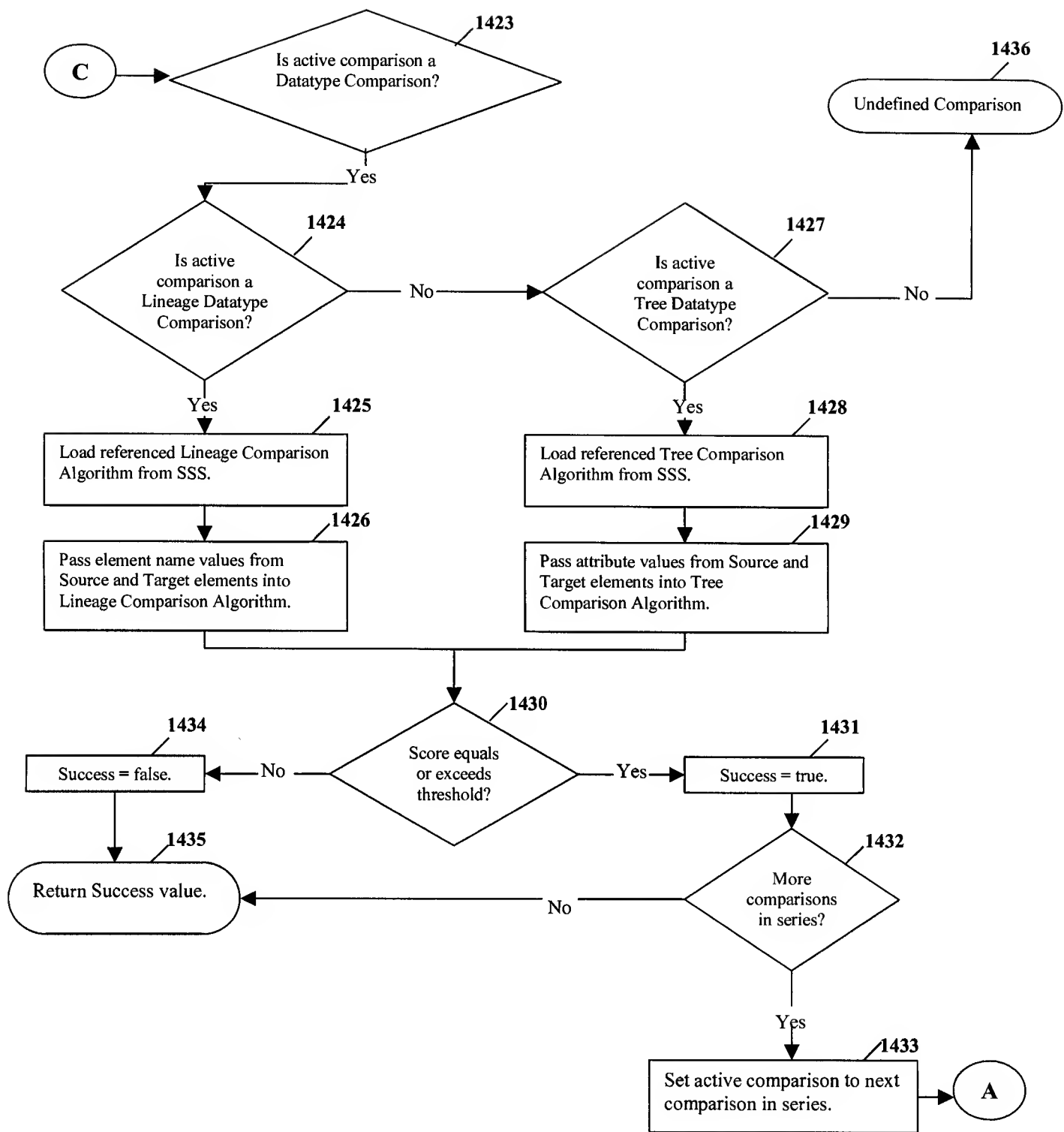


FIG. 14C